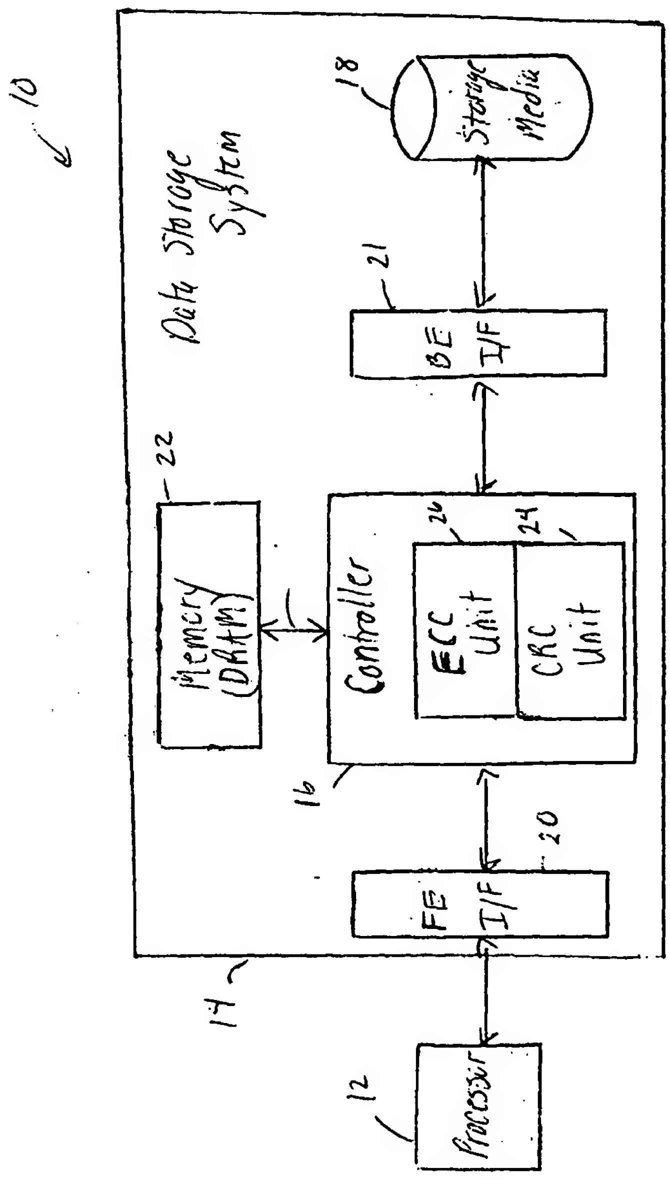

10
Data Storage System
18
Storage Media
21
BE I/F
22
Memory (DRAM)
Controller
26
24
ECC Unit
CRC Unit
16
20
FE I/F
14
12
Processor

FIG. 1

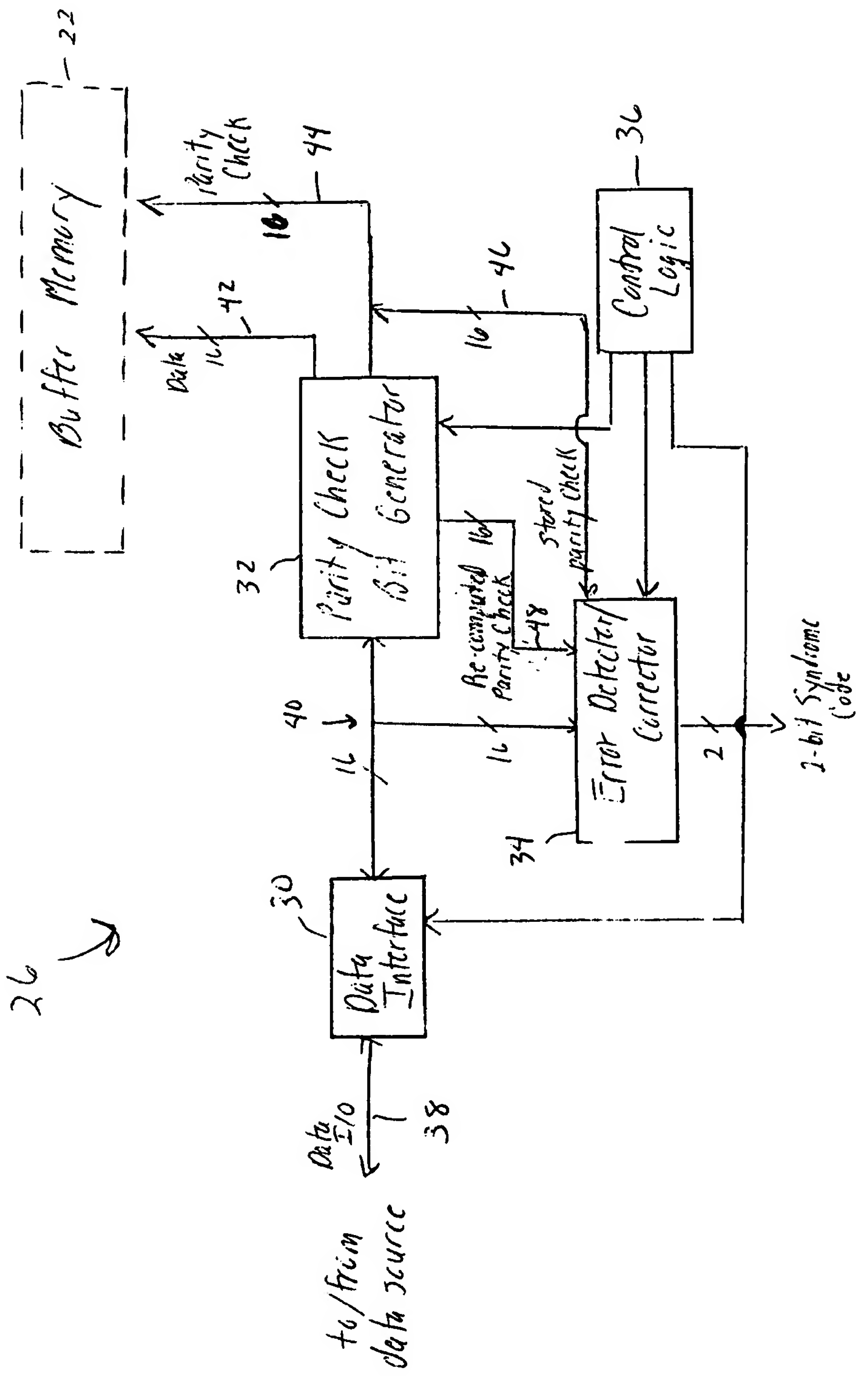

Buffer Memory
22
Parity Check
44
16
Data
16
42
46
16
Control Logic
36
Parity Check Bit Generator
32
Stored parity check
Re-computed parity check
16
48
Error Detector/Corrector
34
40
16
16
2
2-bit Syndrome code
Data Interface
30
26
Data I/O
38
to/from data source

FIG. 2

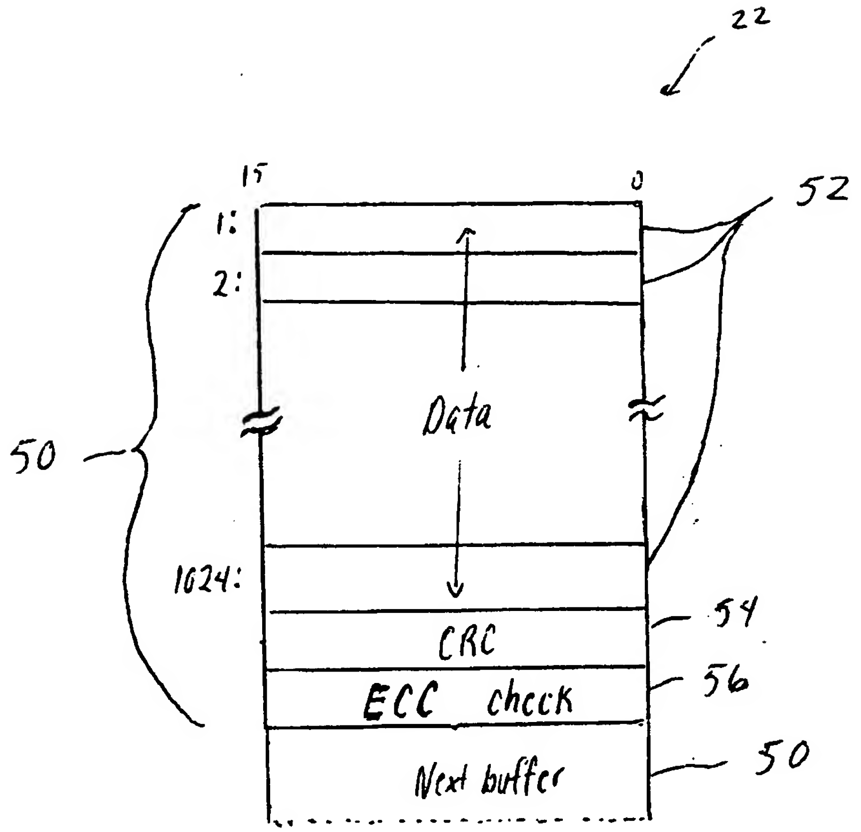
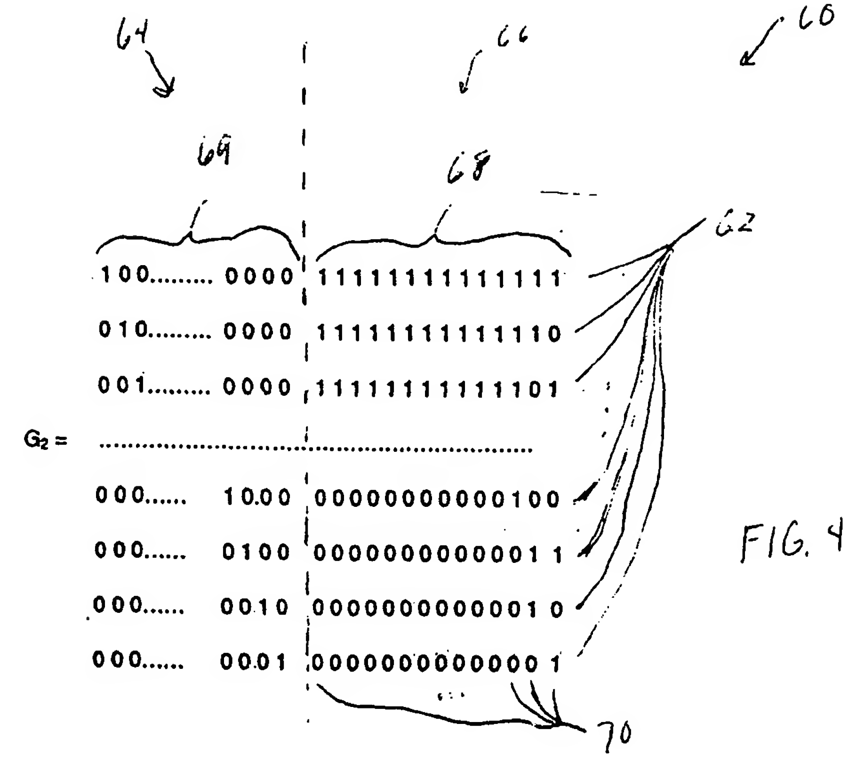
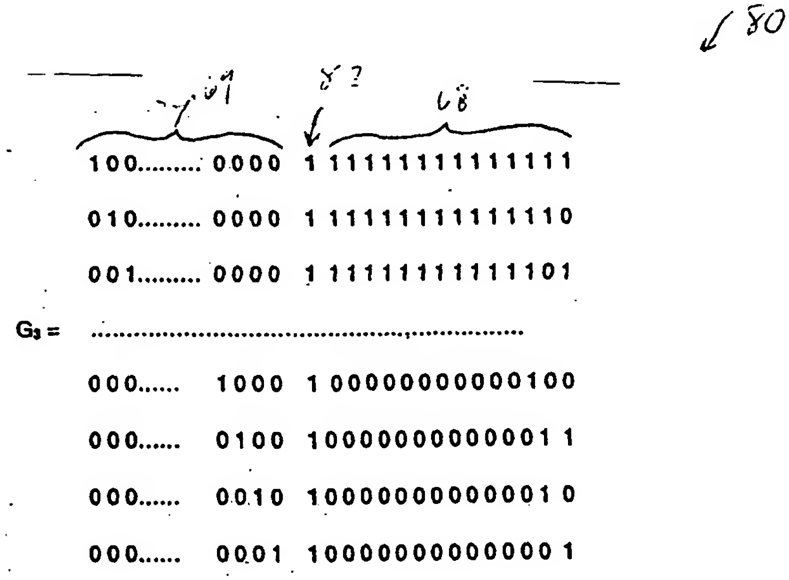


FIG. 3



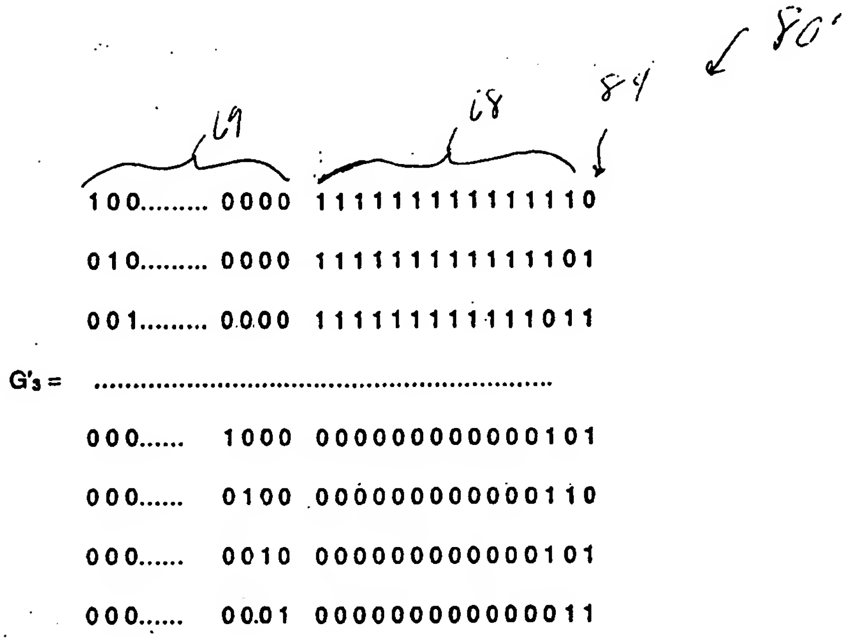
```
       100......... 0000 11111111111111
       010......... 0000 11111111111110
       001......... 0000 11111111111101
G2 =   ..........................................
       000......   10.00 00000000000100
       000......   0100  0000000000001 1
       000......   0010  0000000000001 0
       000......   0001  0000000000000 1
```

FIG. 4



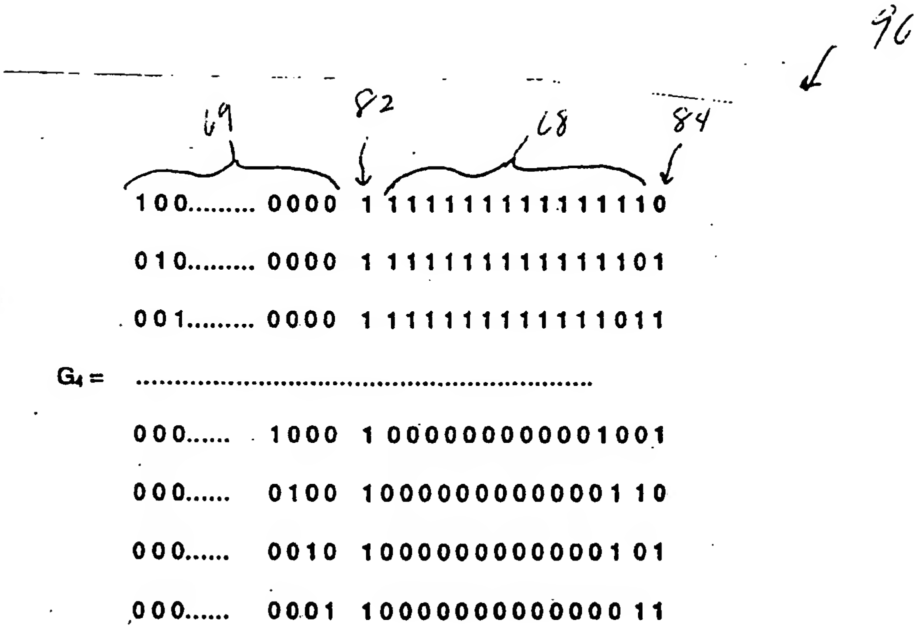
```
       100......... 0000 1 11111111111111
       010......... 0000 1 11111111111110
       001......... 0000 1 11111111111101
G3 =   ..........................................
       000......   1000  1 00000000000100
       000......   0100  10000000000001 1
       000......   0010  10000000000001 0
       000......   0001  10000000000000 1
```

FIG. 5



```
        100......... 0000  11111111111110
        010......... 0000  111111111111101
        001......... 0000  111111111111011
G'3 =   .............................................................
        000......    1000  000000000000101
        000......    0100  000000000000110
        000......    0010  000000000000101
        000......    0001  000000000000011
```

FIG. 6



```
        100......... 0000  1 111111111111110
        010......... 0000  1 111111111111101
        001......... 0000  1 111111111111011
G4 =    .............................................................
        000......    1000  1 000000000001001
        000......    0100  10000000000001 10
        000......    0010  10000000000001 01
        000......    0001  10000000000000 11
```

FIG. 7

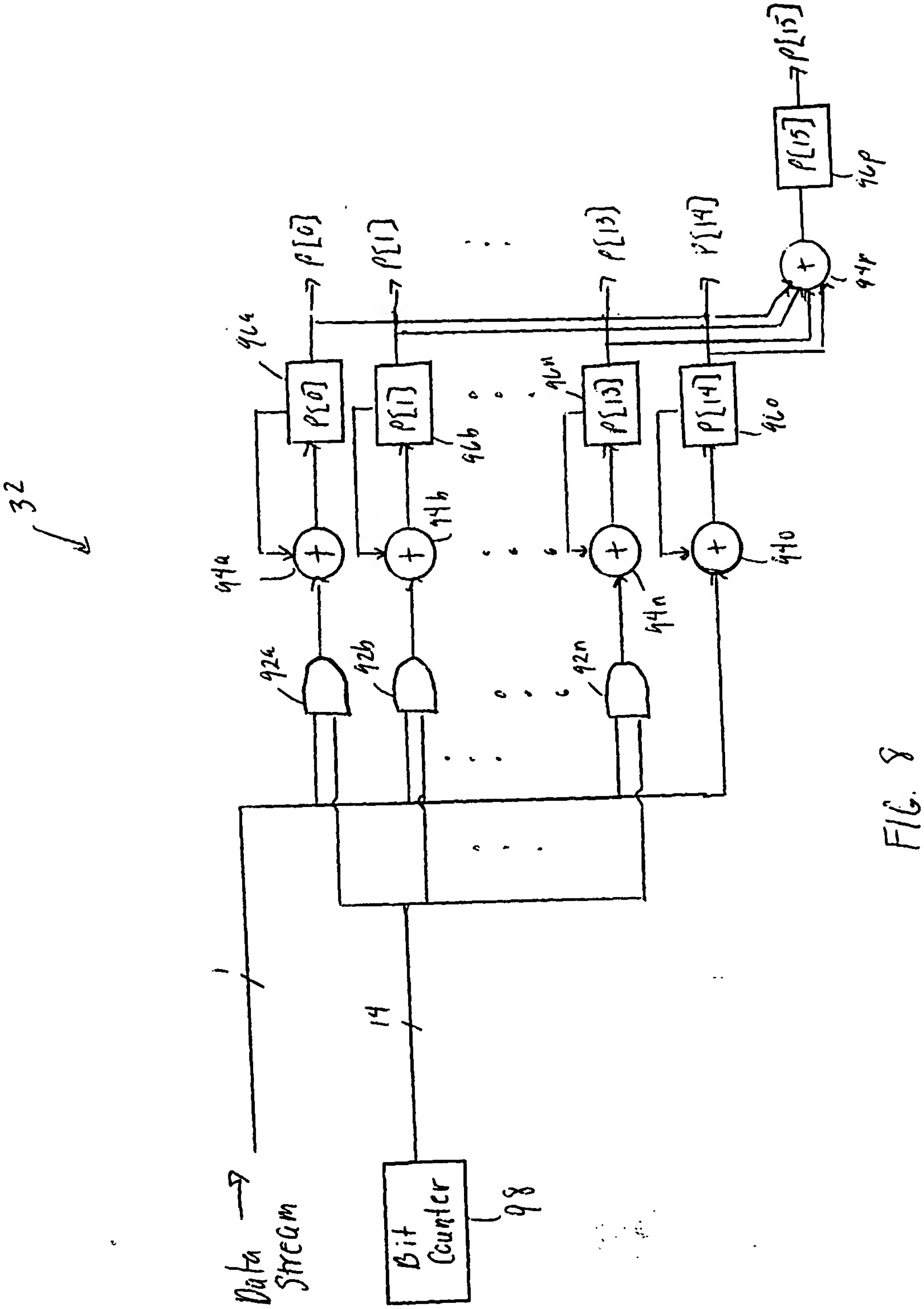

32
Data Stream
Bit Counter
98
1
14
92a
92b
92n
94a
94b
94n
94o
96a
96b
96n
96o
p[0]
p[1]
p[13]
p[14]
p[0]
p[1]
p[13]
p[14]
94r
p[15]
96p
p[15]

FIG. 8

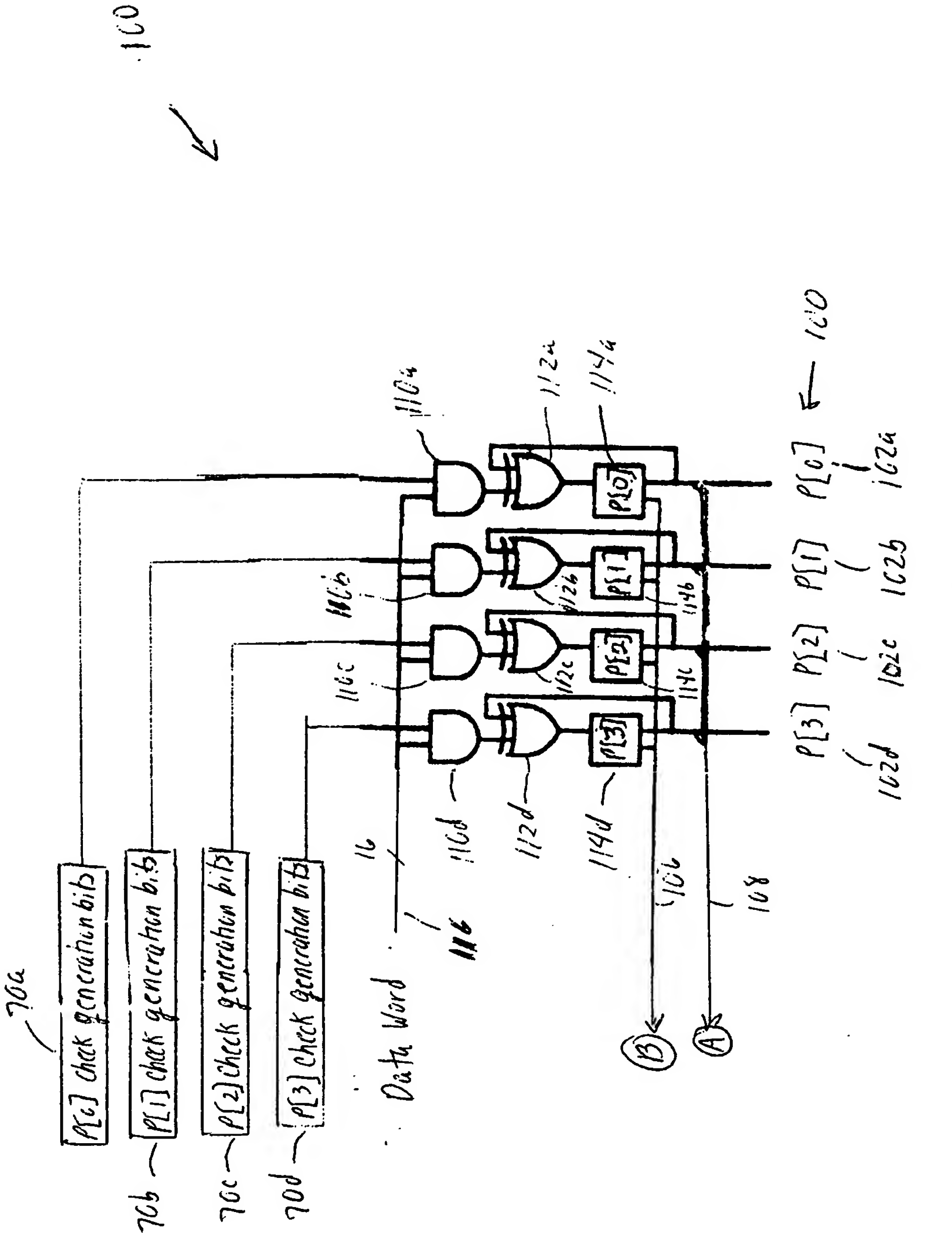

100
110a
112a
114a
P[0]
100
P[0]
102a
P[1]
102b
P[2]
102c
P[3]
102d
P[1]
110b
112b
114b
P[2]
110c
112c
114c
P[3]
110d
112d
114d
116
116
106
108
B
A
P[0] check generation bits
P[1] check generation bits
P[2] check generation bits
P[3] check generation bits
70a
70b
70c
70d
Data Word

FIG. 9A

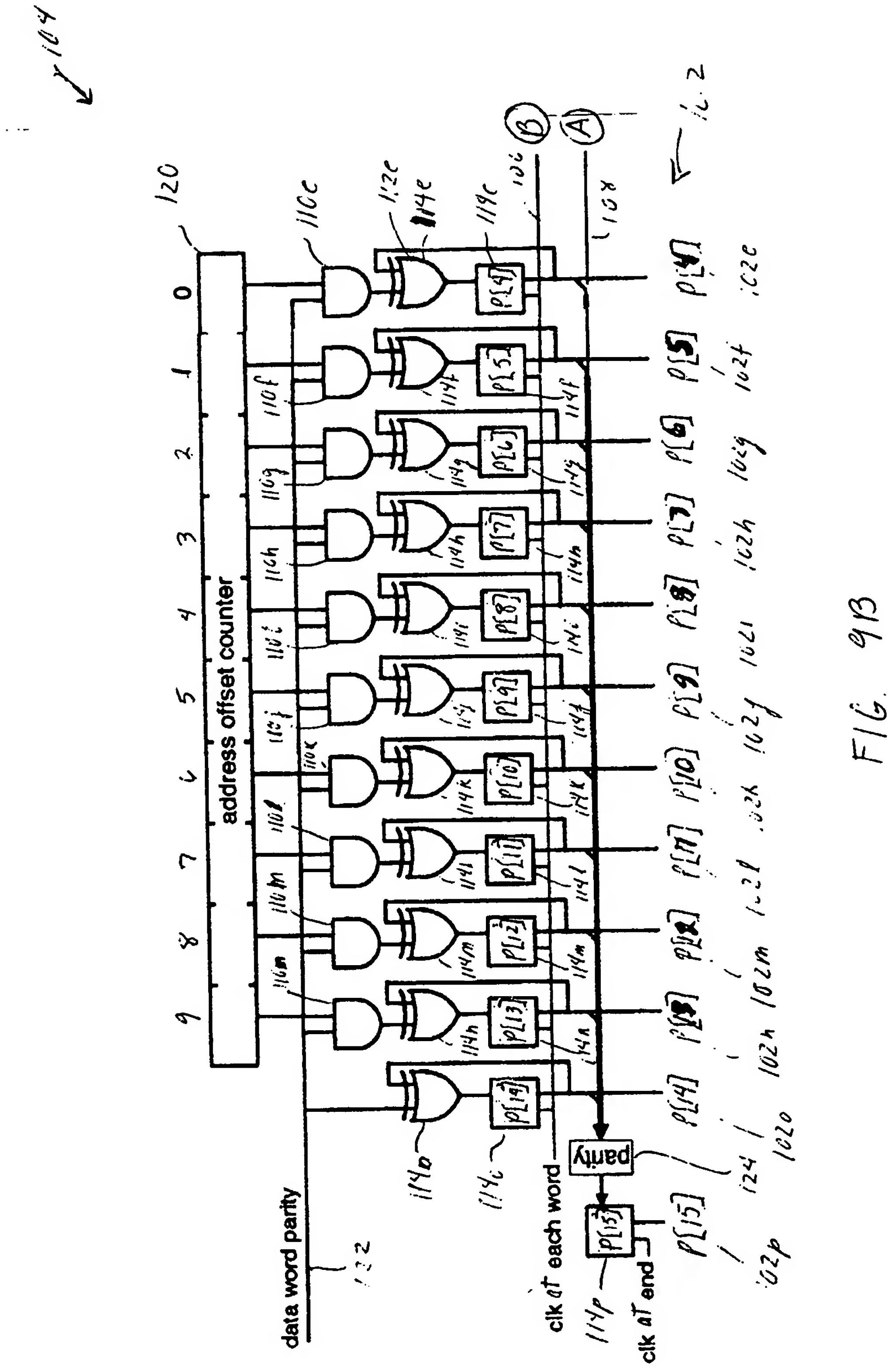

104
120
address offset counter
0
1
2
3
4
5
6
7
8
9
110e
110f
110g
110h
110i
110j
110k
110l
110m
110n
112e
114e
114f
114g
114h
114i
114j
114k
114l
114m
114n
114o
114c
114p
P[4]
P[5]
P[6]
P[7]
P[8]
P[9]
P[10]
P[11]
P[12]
P[13]
P[14]
P[15]
parity
data word parity
clk at each word
clk at end
106
108
B
A
102
102e
102f
102g
102h
102i
102j
102k
102l
102m
102n
102o
102p
122
124

FIG. 9B

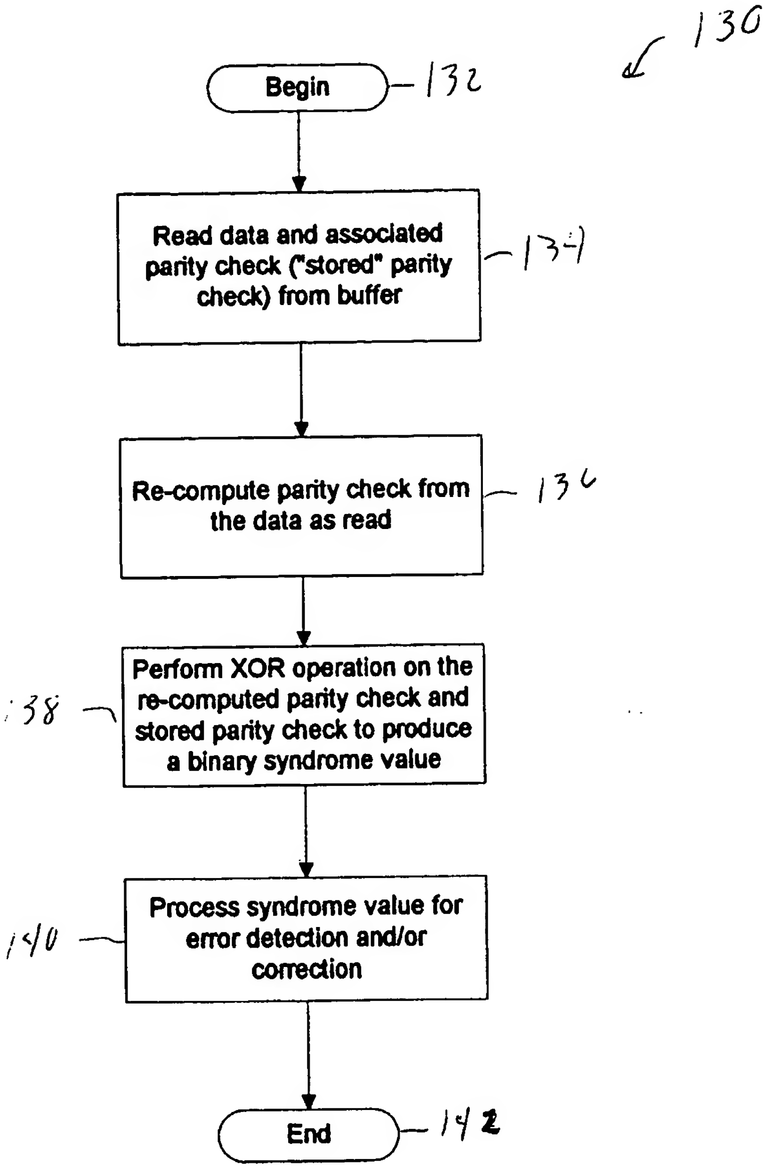

130
Begin
132
Read data and associated parity check ("stored" parity check) from buffer
134
Re-compute parity check from the data as read
136
Perform XOR operation on the re-computed parity check and stored parity check to produce a binary syndrome value
138
Process syndrome value for error detection and/or correction
140
End
142

FIG. 10

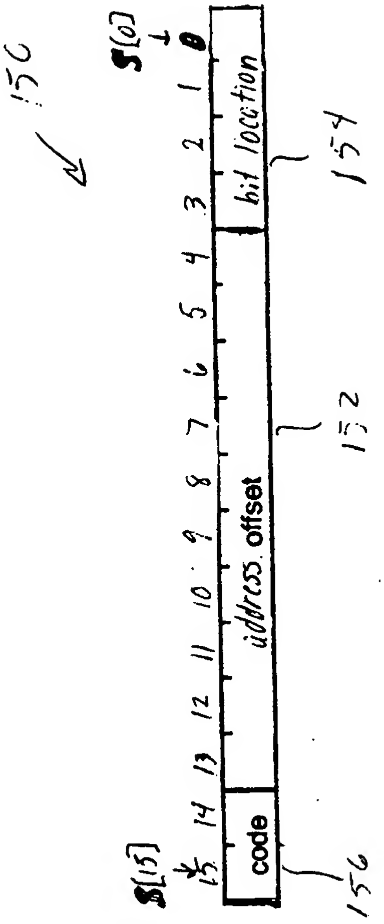
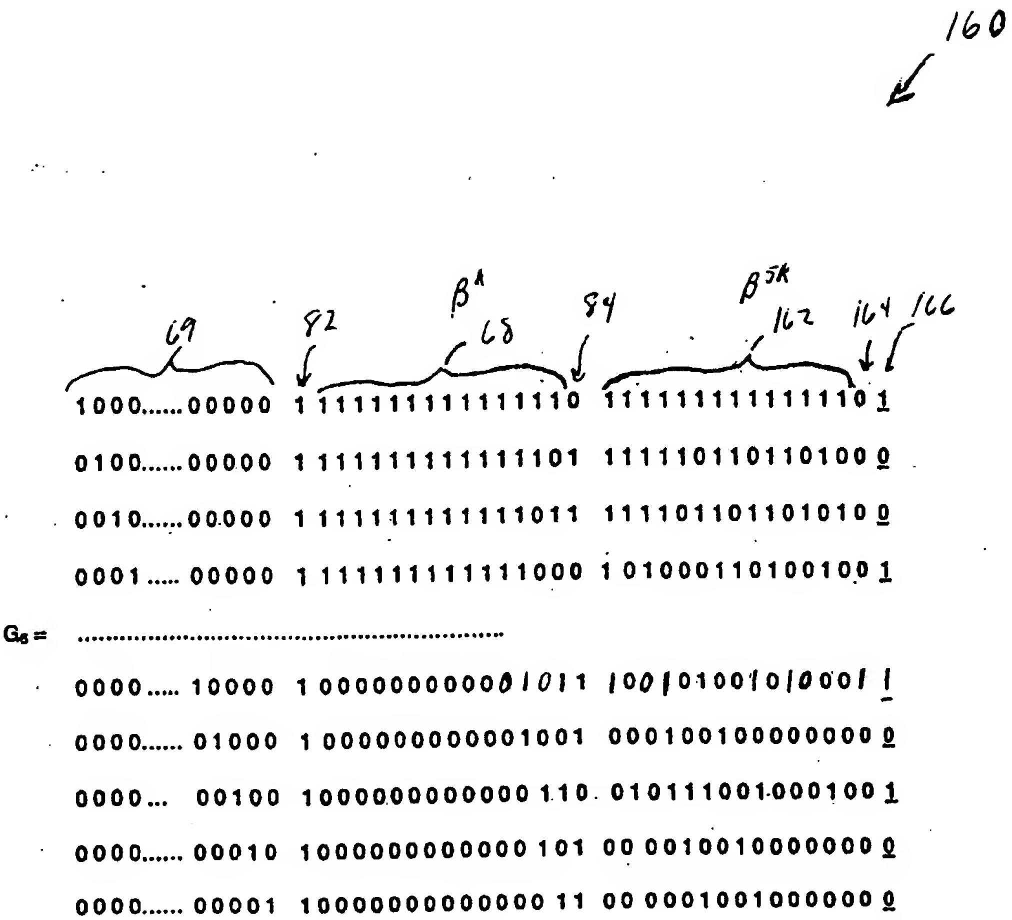
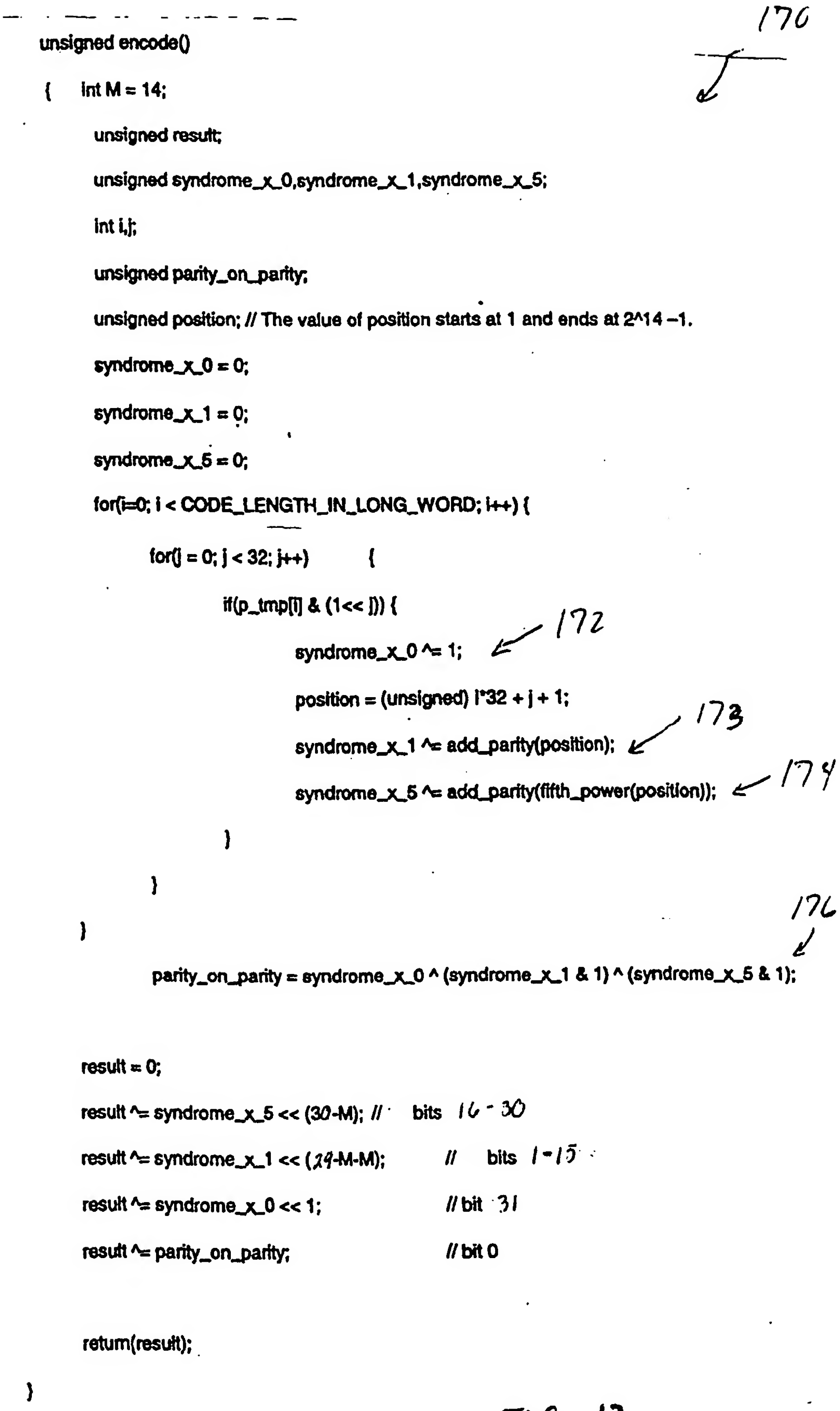

150
S[0]
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
S[15]
bit location
address offset
code
154
152
156

FIG. 11



```
      1000......00000  1 111111111111110  111111111111110 1
      0100......00000  1 111111111111101  111110110110100 0
      0010......00000  1 111111111111011  111101101101010 0
      0001 ..... 00000 1 111111111111000  1 01000110100100 1
G6 =  ..........................................................
      0000 ..... 10000 1 000000000001011  100101001010001 1
      0000...... 01000 1 000000000001001  000100100000000 0
      0000...    00100 10000000000000 110 010111001000100 1
      0000...... 00010 1000000000000 101  00 0010010000000 0
      0000...... 00001 10000000000000 11  00 0001001000000 0
```

FIG. 12

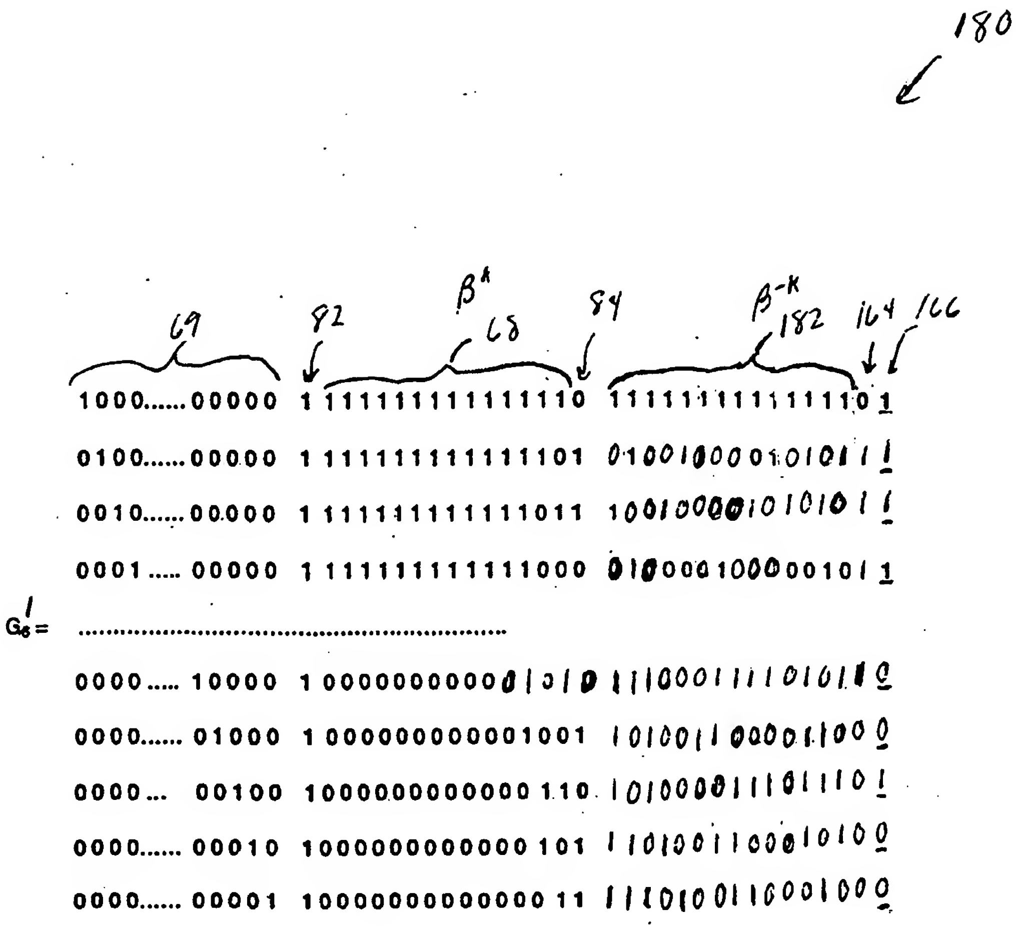


```
unsigned encode()
 {   int M = 14;
        unsigned result;
        unsigned syndrome_x_0,syndrome_x_1,syndrome_x_5;
        int i,j;
        unsigned parity_on_parity;
        unsigned position; // The value of position starts at 1 and ends at 2^14 -1.
        syndrome_x_0 = 0;
        syndrome_x_1 = 0;
        syndrome_x_5 = 0;
        for(i=0; i < CODE_LENGTH_IN_LONG_WORD; i++) {
             for(j = 0; j < 32; j++)       {
                  if(p_tmp[i] & (1<< j)) {
                       syndrome_x_0 ^= 1;
                       position = (unsigned) i*32 + j + 1;
                       syndrome_x_1 ^= add_parity(position);
                       syndrome_x_5 ^= add_parity(fifth_power(position));
                  }
             }
     }
             parity_on_parity = syndrome_x_0 ^ (syndrome_x_1 & 1) ^ (syndrome_x_5 & 1);

     result = 0;
     result ^= syndrome_x_5 << (30-M); //   bits 16 - 30
     result ^= syndrome_x_1 << (29-M-M);      //   bits 1-15
     result ^= syndrome_x_0 << 1;             // bit 31
     result ^= parity_on_parity;              // bit 0

     return(result);
 }
```

FIG. 13



```
       1000......00000  1 111111111111110  111111111111110 1
       0100......00000  1 111111111111101  010010000101011 1
       0010......00000  1 111111111111011  100100001010101 1
       0001 ..... 00000  1 111111111111000  010000100000101 1
G'6 =  ..........................................................
       0000..... 10000  1 000000000001010  111000111101011 0
       0000...... 01000  1 000000000001001  101001100001100 0
       0000...  00100  1000000000000 110  101000011101110 1
       0000...... 00010  1000000000000 101  110100110001010 0
       0000...... 00001  10000000000000 11  111010011000100 0
```

FIG. 14